

CLAIMS

1 1. A method for establishing a location transparent event handler comprising the
2 steps of:

3 establishing a Notifier object in a client application for execution in a first process
4 address space, said Notifier object based upon a Notifier class, said Notifier object

5 having a list of Listener objects to be notified upon an event occurrence;

6 establishing a Listener object in a server application for execution in a second
7 process address space separate from said first process address space, said Listener
8 object based upon a Listener class, said Listener object defining a method to be called
9 upon the occurrence of said event, said Listener object enabled to be callable from said
10 Notifier object; and,

11 generating a Listener object stub for said Listener object, said Listener object
12 stub configured to be added to said list of Listener objects in said Notifier object, said
13 Listener object stub further configured to remotely call said defined method in said
14 Listener object in response to receiving notification of an event from said Notifier object,

15 whereby upon said event occurrence, said Notifier object can traverse said list of
16 Listener objects and can notify said Listener object stub of said event occurrence
17 thereby creating a remote call to said defined method in said Listener object.

1 2. The method of claim 1, wherein said Notifier and Listener classes are Java
2 classes and said first and second process address spaces are in first and second Java
3 Virtual Machines, respectively.

1 3. The method of claim 2, wherein said generating step comprises the steps of:

2 RMI compiling said Listener class, said RMI compilation generating said Listener
3 object stub; and,

4 registering said Listener object with an RMI Registry, said RMI Registry
5 executing in a third Java Virtual Machine,

6 said Notifier object retrieving a reference to said registered Listener object from
7 said RMI Registry upon said addition said Listener object stub to said list of Listener
8 objects,

9 said Listener object stub remotely calling said defined method in said Listener
10 object through said retrieved reference upon receiving notification of an event from said
11 Notifier object.

1 4. A method for performing location transparent event handling comprising the
2 steps of:

3 creating an instance of a Notifier class in a first process address space, said
4 Notifier instance having a list of Listener objects to be notified upon an event
5 occurrence;

6 creating an instance of a Listener class in a second process address space, said
7 Listener instance having a method to be called upon the occurrence of said event, said
8 Listener instance enabled to be callable from said Notifier instance;

9 inserting a Listener object stub in said list of Listener objects in said Notifier
10 instance in said first process address space, said Listener object stub configured to
11 remotely call said defined method in said Listener instance;

12 receiving an event occurrence in said Notifier instance; and,

13 responsive to receiving said event occurrence, traversing said list of Listener
14 objects, passing said event to said Listener object stub, creating in said Listener object
15 stub a remote call to said defined method in said Listener instance, and executing said
16 defined method in said Listener instance.

1 5. The method of claim 4, wherein said Notifier and Listener classes are Java
2 classes and said first and second process address spaces are in first and second Java
3 Virtual Machines, respectively.

1 6. The method of claim 5, wherein said Listener object stub is generated in an RMI
2 compilation process.

cont

7. The method of claim 6, wherein said inserting step further comprises the step of:
registering said Listener instance with an RMI Registry, said RMI Registry
executing in a third Java Virtual Machine,

Sub²
A₁ said Notifier instance retrieving a reference to said registered Listener instance
5 from said RMI Registry upon inserting said Listener object stub to said list of Listener
6 objects.

Sub³
A₂ 8. The method of claim 7, wherein said step of creating in said Listener object stub
remotely calls said defined method in said Listener instance through said retrieved
reference upon receiving said event from said Notifier instance.

A₃ 9. A machine readable storage, having stored thereon a computer program having
a plurality of code sections for establishing a location transparent event handler, said
code sections executable by a machine for causing the machine to perform the steps
of:

A₄ establishing a Notifier object in a client application for execution in a first process
address space, said Notifier object based upon a Notifer class, said Notifier object
having a list of Listener objects to be notified upon an event occurrence;

A₅ establishing a Listener object in a server application for execution in a second
process address space separate from said first process address space, said Listener
object based upon a Listener class, said Listener object defining a method to be called
upon the occurrence of said event, said Listener object enabled to be callable from said
Notifier object; and,

A₆ generating a Listener object stub for said Listener object, said Listener object
stub configured to be added to said list of Listener objects in said Notifier object, said

15 Listener object stub further configured to remotely call said defined method in said
16 Listener object in response to receiving notification of an event from said Notifier object,
17 whereby upon said event occurrence, said Notifier object can traverse said list of
18 Listener objects and can notify said Listener object stub of said event occurrence
19 thereby creating a remote call to said defined method in said Listener object.

SJ
A/

10. The machine readable storage of claim 9, wherein said Notifier and Listener
2 classes are Java classes and said first and second process address spaces are in first
3 and second Java Virtual Machines, respectively.

11. The machine readable storage of claim 10, wherein said generating step
comprises the steps of:

RMI compiling said Listener class, said RMI compilation generating said Listener
object stub; and

registering said Listener object with an RMI Registry, said RMI Registry
executing in a third Java Virtual Machine,

said Notifier object retrieving a reference to said registered Listener object from
said RMI Registry upon said addition said Listener object stub to said list of Listener
objects,

10 said Listener object stub remotely calling said defined method in said Listener
11 object through said retrieved reference upon receiving notification of an event from said
12 Notifier object.

1 12. A machine readable storage, having stored thereon a computer program having
2 a plurality of code sections for performing location transparent event handling, said
3 code sections executable by a machine for causing the machine to perform the steps
4 of:

5 creating an instance of a Notifier class in a first process address space, said

6 Notifier instance having a list of Listener objects to be notified upon an event
7 occurrence;

8 creating an instance of a Listener class in a second process address space, said
9 Listener instance defining a method to be called upon the occurrence of said event,
10 said Listener instance enabled to be callable from said Notifier instance;

11 inserting a Listener object stub in said list of Listener objects in said Notifier
12 instance in said first process address space, said Listener object stub configured to
13 remotely call said defined method in said Listener instance;

14 receiving an event occurrence in said Notifier instance; and,

15 responsive to receiving said event occurrence, traversing said list of Listener
objects, passing said event to said Listener object stub, creating in said Listener object
stub a remote call to said defined method in said Listener instance, and executing said
defined method in said Listener instance.

13. The machine readable storage of claim 12, wherein said Notifier and Listener
classes are Java classes and said first and second process address spaces are in first
and second Java Virtual Machines, respectively.

14. The machine readable storage of claim 13, wherein said Listener object stub is
2 generated in an RMI compilation process.

15. The machine readable storage of claim 14, wherein said inserting step further
2 comprises the step of:

3 registering said Listener instance with an RMI Registry, said RMI Registry
4 executing in a third Java Virtual Machine,

5 said Notifier object retrieving a reference to said registered Listener object from
6 said RMI Registry upon inserting said Listener object stub to said list of Listener
7 objects.

16. The machine readable storage of claim 15, wherein said step of creating in said
Listener object stub remotely calls said defined method in said Listener instance
through said retrieved reference upon receiving said event from said Notifier instance.